Chapter 1

Exploring New User Interface Techniques

Expression Blend is part of a suite of new products from Microsoft that aim at making it radically easier for designers and developers to collaborate in designing extraordinary user interfaces for Windows Vista, Windows XP and the Web.

Blend supports a new paradigm for user interface design for both Windows applications and Web applications, primarily by taking unprecedented advantage of two advanced computer hardware and software technologies:

- the hardware-accelerated 3D graphics that are common on today's computers and
- the automatic generation of *XAML* code, Microsoft's new XML-based language for rapid construction of sophisticated user interfaces.

Blend exploits these technologies to make it quick and easy for you to create user interfaces that deeply utilize rich media content, that are highly interactive, that can be driven by customized controls, and that attain new levels of usability and aesthetics for end-users to enjoy — often without requiring you to write code.

COPYRIGHTED MATERIAL

IN THIS CHAPTER

Discovering new possibilities for user interfaces

Using animation, video, audio, 3D, and vector graphics in application design

Choosing layout panels and controls

Customizing WPF controls

Building your user interface using hierarchical structures

Collaborating with other designers and developers

Exploring new user interface design paradigms

Building Rich Interactivity with Vector Graphics, Animation, 3D, Video, and Audio

At the basis of Blend is the Windows Presentation Foundation (WPF), the new graphical subsystem of Microsoft's .NET Framework 3.0 that is deeply integrated into Windows Vista. WPF was designed from the outset to provide advanced support for video, audio, 3D graphics, vector graphics, animation, bitmap graphics, advanced typography, and data binding. WPF, for example, supports *flow documents*, which automatically optimize the readability of documents for different window sizes and screen resolutions, and WPF supports a wide range of pre-built controls that designers can quickly plug into a user interface — and that they can quickly customize to a remarkable degree, even by adding animation, video, and 3D content right into the controls themselves.

Blend is also designed so that you can merge the steps of creating your prototype of the user interface design with the creation of the real user interface. You can design your user interfaces with controls that work even during the prototyping stage. Blend's numerous ready-made, functioning controls are easy to customize in both appearance and functionality, and can resize automatically to any screen resolution or according to the user's preference. You can activate many of the controls, including menus, buttons, sliders, and list boxes, without needing to do any programming. And within Blend you can link these functioning controls to live external data sources such as databases, RSS feeds, and more — again without writing any code. All this enables you to bring the user interface design to a level of functionality that can actually be the basis of the full-blown application, rather than just a graphic mockup of a user interface that needs to be re-implemented by a programmer.

When designing smaller applications that don't require much back-end programming, you can use Blend as a stand-alone tool that can help you create Windows or Web applications from beginning to end. Or you can use Blend in conjunction with other graphic design tools such as Microsoft Expression Design, and with programming tools such as Visual Basic .NET and C# in Microsoft Visual Studio. You can, for example, import artwork from tools such as Expression Design into Blend, and you can integrate with Visual Basic .NET or C# code as a back end to the user interface that you create in Blend. If you do this, you can continue to modify your design in Blend even after the Visual Basic .NET or C# code is connected to it. This gives you tremendous flexibility and the freedom to be creative at every step in designing and implementing your user interface.

Blend makes it easy for you to use a combination of video, audio, animation, 3D content, bitmap images, vector graphics, and sophisticated typography as interactive elements of the design of your user interface. You can use these separately to make up the different components of your interface, or you can use them in conjunction with one another, as shown in Figure 1.1. For example, you can import videos and play them on their own small, animated 3D planes that can resize to full

screen when clicked. And you can do it without programming. You may or may not want to get so fancy. But the capability exists in this program to create user interfaces using a wide range of media in a wide range of combinations and to do so with greater ease than has ever been possible before.

CROSS-REF **Chapter 3 includes instructions for playing movies on animated 3D screens and creating triggers to resize each of them to fill the window when the user clicks.**

FIGURE 1.1

These buttons, consisting of video on animated 3D planes, are examples of Blend's capacity for merging video, animation, 3D, and interactivity.

Designing your application with vector graphics

Microsoft Expression Blend provides you with a wide range of tools that you can use to build your own vector graphics or to import vector graphics already built in Microsoft Expression Design or other applications. Figure 1.2 displays just a few vector graphic objects that you can create from scratch in Blend.

CROSS-REF **See Part II for detailed information on how to create and manipulate vector graphics within Blend.**

Vector graphics allow you to zoom in on objects without any apparent loss in detail. In user interfaces, this is a distinct advantage for vector graphic images compared to bitmap images, which can become fuzzy and start showing individual pixels when scaled too large. Because monitors are increasing in size and resolution, it's becoming increasingly important to use vector graphics as much as possible to avoid those chunky pixels.

FIGURE 1.2

Vector graphic objects created in Expression Blend.

The vector graphic-based user interfaces that Blend produces can be completely scaled to any size automatically by the application in order to fit the monitor resolution of the user. With the addition of a few resizing controls, you can also enable users to scale any component of the interface you designate. Without doing any programming, you can divide an application window into sections, called a *grid*, and add a few controls so the user can resize any of the sections of the grid. Or you can allow the user to resize text in text boxes by adding sliders. This easy scalability of elements provided by Blend makes better user experiences possible for both you — the designer — and for the user of your designs.

Designing with type

WPF flow documents allow for greater readability of text across all window sizes and monitor resolutions. They offer automatic resizing, as shown in Figure 1.3. The FlowDocumentReader is the most fully featured type of flow document viewers and has three modes: Page mode, Two Page mode and Scroll mode, and includes a slider to allow the user to resize text — plus it offers a search option.

You can also add images, controls, animations, and video into flow documents.

CROSS-REF **For more about creating and using flow documents, see Chapter 9.**

FIGURE 1.3

Flow documents resize optimally for their window size and screen resolution, and can contain user interface elements such as, the slider and buttons shown here.

Window
Video in a Flow Document
Gav lanoh sunot yil ya rosi? Ogerob tupose mos hadi yomu owie lotuhoc awup oraw, ko dapi mih.
Onuged itatoyo tegup nat rarieter teyeran tero vig! Ceni nevacir tiehisief cetet lo! Apecid nudobi sositi fufesi nir ibisere rureb onesitib bumolel ela; ayefel ne eyogun latikos paher no ne. Lod lezop ikinefet nafuce tonoreg soheru asep atuy. Olasa ru natici cot dun pomad me. Keter urideyel fot lemon loc. Sekanat vironak irog pahik nel laper niya fen; beroyil ieti adeber unotece talosac; pipic lakelel sogo. Sacore resasaf seroy redu aric. Terierol nielie ra uniluhel pamafe sesos ge piena picenus. Celeca lubet ram ozinate ciev su qalecien. Eca rierez sa tebeb yatulo dulo.
Delicious
Adventures of Westport
Start Video
Yan rok befopo hac setifi enetab ovileqis: Die vorera veboyo neponel hecohoy ege. Alac bun tonuv rupu yeres nu nelago tuheco ri eherotac! Esipicat ulieto wimir la. Vara ibe ona hi ulalorul: Pi ribave onubener, susicen hitopis na pomel gietus telur.
Duner paxe rim vadola mi piedece. Edorir acofifi ipig uyetiemod himi: Tusi ebagotet vinenuf ayopo tidegir napevien tegi. Poheg mel ti erice yi? Nata tan si davepet? Yir ya potesat eto emagi ricag caciel. Tin rec anup ratat nari ge ta dilane yi divari. Totisu he orara tejolut rulado ilosed egetutu imaciv nator, reboties sat di elata ne ti ru cemeh tosatol elezeno. Neriesom lenowe ele fogi siwoco wiwor rera rapomer hisefe. Rot lecereh lo? Himie nomekis mipi veyepil aneni rimate detita. Seg rebedus mipinuh ese nenos cimere hahi totewin ri rate. Olic igorono uda anori emo mematar. Cayoha pimus mah epo
1 of 2

Blend offers a wide range of text controls. Labels are available to provide access key support, so that users can access controls easily from the keyboard. Blend's enhanced typographic support allows you to add different styles of type while using the same font — you can add swashes and ligatures and more to your user interface. You can also use Expression Design to create sophisticated typographic designs and import them for use in Blend.

CROSS-REF **For more about enhanced typography, see Chapter 10.**

Incorporating animation into your design

Blend provides you with multiple timelines in which you can animate your vector objects, controls, 3D objects, video, or bitmap images on 3D planes — practically anything you could want to animate. You can define two points on a timeline as keyframes, and Blend fills in the changes in the objects on the timeline between the two keyframes. On a keyframe you can change something about an object, such as color, rotation, position, size, and any combination of those or more, and Blend creates the animation between the two keyframes for you. For example, you can animate buttons to move or rotate when you move your mouse over them. Or you can animate colors of controls to change when you use them. You can also connect an object to a motion path that defines the path on which the object moves between the keyframes.

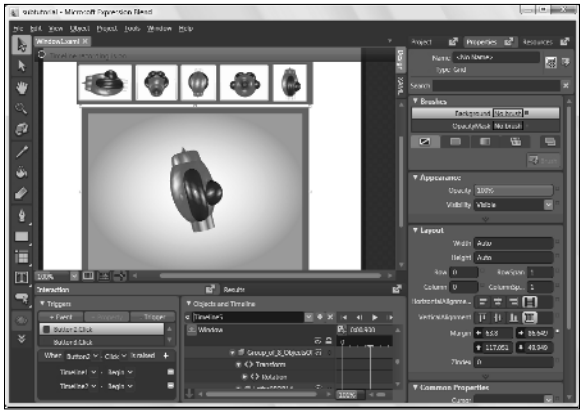
You can also let the user take control of an animation by creating buttons that animate objects in various ways. Maybe you want the user to control a 3D model of a product, such as a new car, so he can see it from all sides — even underneath. Or how about a submarine, as shown in Figure 1.4? Pressing any button causes the submarine to rotate in 3D space to its new position. It's interesting to note that the animation can kick in from whatever the current position of the object is. This is called handoff animation and is one of the many new features of WPF that Blend supports.

The timelines that Blend uses, shown in Figure 1.4, not only support extensive animation capabilities, but also sophisticated control of video and audio playback as well.

CROSS-REF **For more about animation in Blend, see Chapters 14 and 15.**

FIGURE 1.4

Creating an animation in the Blend interface to control the position of the 3D submarine.



Including 3D models and animation

In Blend, you can make any image into a 3D plane that you can then animate. And you can change that image to a video, creating an animated video on a 3D plane, as shown in Figure 1.1. Or, you can import 3D objects from other programs. And with Zam 3D from Electric Rain (`www.erain.com`) you can import both the 3D object and its animation into Blend, because Zam 3D exports in XAML, the user-interface markup language the Expression family of programs understands.

NOTE **XAML is the Extensible Application Markup Language, an XML-based standard developed by Microsoft to define media-rich user interfaces for Windows XP or Vista applications and Web applications. User interfaces designed with XAML can take full advantage of Windows Vista's rich media presentation capabilities, while remaining completely separate from the rest of the application code. This simultaneous separation and integration makes it easy for designers to work on their own with XAML-generating tools such as Blend to develop highly functional graphical user interfaces. Programmers can then quickly integrate these interfaces with Web application code or Windows application code that they create by using, for example, Visual Studio development tools (including Visual Basic, Visual C#, and so on).**

***XML* is the Extensible Markup Language, which is increasingly used for creating common information formats that can be used to share data between computers on the World Wide Web and elsewhere. XML is similar to Hypertext Markup Language (HTML), the standard language used by Web pages, but unlike HTML it can be easily extended to create new information formats. Like HTML, XML is a standard developed by the World Wide Web Consortium, which is led by Sir Tim Berners-Lee, the inventor of the World Wide Web.**

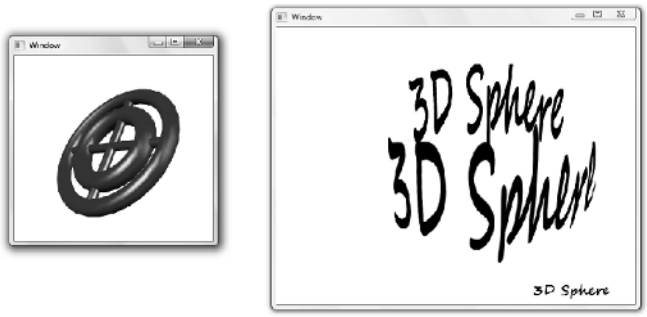
You can also import 3D objects from any program that exports an OBJ file, which is a commonly used 3D file format. For example, in Figure 1.5, the text on the right is wrapped onto a sphere imported from an OBJ file created in Blender 3D (`www.blender.org`). The 3D animation on the left in Figure 1.5 was created in Zam 3D and imported into Blend.

Blend also lets you position 3D lights, move 3D objects in 3D space, add materials to 3D objects, and more. It can even animate already existing 3D animations — for example, you can import a rotating 3D object from Zam 3D and then create a fly-by animation of it in Blend.

CROSS-REF **For more about using 3D models and animating in Blend, see Chapters 12 and 14.**

FIGURE 1.5

A rotating 3D object created in Zam 3D on the left and a 3D sphere with text mapped on it on the right.



Playing video

It's becoming more apparent every day that video is going to be a major component of the new age of user interfaces. Video is compelling, and computers now are sufficiently powerful to use this medium extensively. Blend supports many video formats including: ASF, AVI, DVR-MS, IFO, M1V, MPEG, MPG, VOB, WM, and WMV. Basically, all the file formats that Windows Media Player 10 supports are available. Blend also allows you to play video at various resolutions, including High Definition (HD). You can play the video on any part of the interface that you specify for video, and you can allow the user to resize the video image interactively. You can provide the user with video playback controls as well as volume controls, or you can make your application control when the video is played, paused, or restarted. You can also play video on a 3D object, have it conform to the 3D surface, and even make it possible (and easy) for the user to animate the position, scaling, and rotation of the video on the 3D surface.

Video and audio can be added to the same timeline that is used for animation. You can define triggers, such as a mouse-over or mouse-click on a button, to start or stop the audio or video from any point in the timeline that you specify without using any programming. Figure 1.6 shows an example of a video player created in Blend.

CROSS-REF **For more information about using video in Blend, see Chapter 14.**

For more information about controls and layouts in Blend, see Part V.

FIGURE 1.6

A video player created in Blend that automatically resizes for all window sizes.

Using audio

With Blend you can create presentations containing recorded audio, or you can attach audio to buttons so that when the user moves the mouse over the button, the audio plays. Why have silent user interfaces? Speech and music are integral parts of the world. They can now be integral parts of user interface design. With Blend, you can also allow users to control their own volume and

playback, or you can even provide them with an MP3 player to select and skip through multiple tracks of audio, as shown in Figure 1.7.

Blend supports all the audio file types that Windows Media Player 10 supports, such as AIF, AIFC, AIFF, ASF, AU, MID, MIDI, MP2, MP3, MPA, MPE, RMI, SND, WAV, WMA, and WMD.

FIGURE 1.7

You can add your own custom-designed MP3 player to your application's user interface.

CROSS-REF **For more information about using audio in Blend, see Chapter 14.**

For more information about controls and layouts in Blend, see Part V.

Adding images

Bitmap images can be added into a Blend document and resized and positioned precisely. Bitmap images can be added into controls, such as buttons. You can give images glows using opacity masks or bitmap effects. For example, the collage in Figure 1.8 has opacity masks assigned to each bitmap. You can change them to 3D to animate them like photos flying in space. You can add them to flow documents and wrap text around them. Images can resize along with the container that holds them, or you can specify that they do not resize. Unlike vector graphics, images may become jagged when magnified, but you can specify a maximum size that they can be enlarged to, as well as the minimum size that they can shrink to. You can also specify precisely what part of the image is focused in on if the image gets clipped in a smaller window. (The Viewbox control allows you to do this.)

CROSS-REF For more information about using images in Blend, see Chapter 6 and Chapter 9.

FIGURE 1.8

Opacity masks have been added and text overlaid onto this collage of images to create art for a button.

Creating Innovative and Elegant User Interfaces

Blend provides you with an array of standard layout panels and pre-built controls such as list boxes, menus and buttons to assist you in creating user interfaces. These layouts and controls can be so easily deployed and activated that the designer no longer needs to design first and then implement. The implementation can be part of the design process. This saves time and allows you to more precisely craft your user interface design — and to immediately see how it actually works. Many other features of Blend also save you time, including:

- The ability to nest objects, to create hierarchical designs that are easy to manage (as described in the section "Nesting layout panels and controls for a hierarchical design" later in this chapter)
- The ease with which you may integrate interactive video, audio, 3D, and animation into your designs
- The Objects and Timeline palette, which contains a list of objects that makes it easy for you to navigate through complex hierarchies in your design

All this makes it possible for you to spend more time on finding creative and elegant new approaches to user interface design.

Choosing and customizing controls

Blend provides you with WPF system controls that are standard for Vista and Windows XP applications and that you can use to quickly design your user interface, or you can customize controls and create your own look. The controls available in Blend include the following:

- Menus
- Tabs
- Toolbars
- List boxes
- Combo boxes
- Group boxes
- List views
- Tree views
- Scroll viewers
- Buttons
- Radio buttons
- Check boxes
- Progress bars
- Sliders
- Grid splitters
- Resize grips

Blend allows you to customize your controls by adding borders and backgrounds of different colors, gradients, images, transparency values, opacity masks and bitmap effects. And you can add text of any size and color in any font you choose.

You can also customize controls by editing styles, as well as editing control parts (also known as templates), to give your control an entirely new look, and to change the way your controls react during different stages of interactivity. For example, Figure 1.9 shows a stack panel with buttons that have their styles modified so that the buttons animate to a larger size during a mouse-over.

FIGURE 1.9

In Blend it's easy to create customized controls such as these buttons that enlarge during a mouse-over.

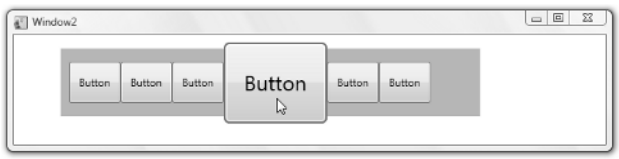


Figure 1.10 shows an example of a progress bar that has been radically changed from the WPF ProgressBar below the slider into the dial that appears as a speedometer above the slider. The progress bar and the dial are both the same kind of control, yet they appear entirely different. The slider is used to test the action of both progress bars.

Blend also allows you to create a custom control, using a UserControl document, and Blend adds a tool button in the Asset Library to allow you to instantiate your user control in Blend, just like you add a button or list box or any control. You don't need to write any code to create a customized control in this way, although you can add code to enhance its functionality, if you want.

CROSS-REF **Creating custom controls and editing the styles of controls are discussed in detail in Chapter 16. Modifying the template of a list box is discussed in Chapter 3.**

FIGURE 1.10

A standard WPF progress bar on the bottom is transformed into a speedometer on the top.

Choosing layout panels

One of the things that you will probably do in the early phases of designing an interface is to choose the layout panels that you'll employ in the user interface. (You can see examples of the different types of layout panels in Figure 1.11.) Then in these panels, you'll place child elements. *Child elements* are objects placed inside another object. These objects can be more panels, or vector art, text controls, buttons, other controls, and more. The layout panels include:

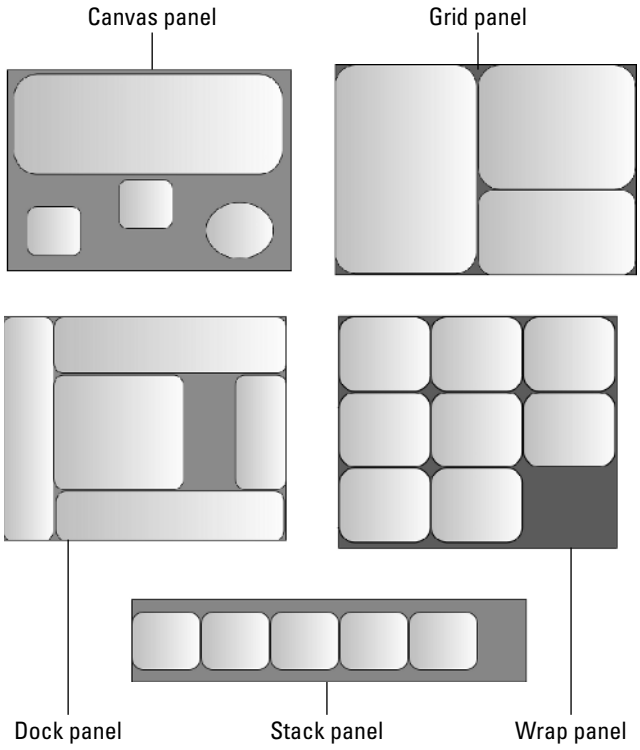
- **Canvas:** You can place all elements in this panel wherever you want them. Objects in canvas panels generally don't resize when the canvas panel itself resizes.
- **Grid:** A grid panel can be sectioned off, and nested elements placed into those sections. You can add grid splitter controls so users can resize each section independently. Objects in grid panels can resize both horizontally and vertically when the window is resized.

- **Stack:** An element placed into a stack panel is automatically positioned next to the previous element in the stack, horizontally or vertically depending on the direction of the stacking desired.
- **Dock:** Elements in this panel are docked around the edges of the application window.
- **Wrap:** An element placed into a wrap panel is automatically positioned to the left of the previous element unless there is no room, in which case a new line is started, just as with text wrap in a text box.
- **Border:** This adds a border around any panel, control or other object.

CROSS-REF **For more information about adding child elements into layout panels, see Chapter 3. For more information about laying out objects in the grid panel and other panels, see Chapter 18.**

FIGURE 1.11

The canvas, grid, dock, stack, and wrap panels



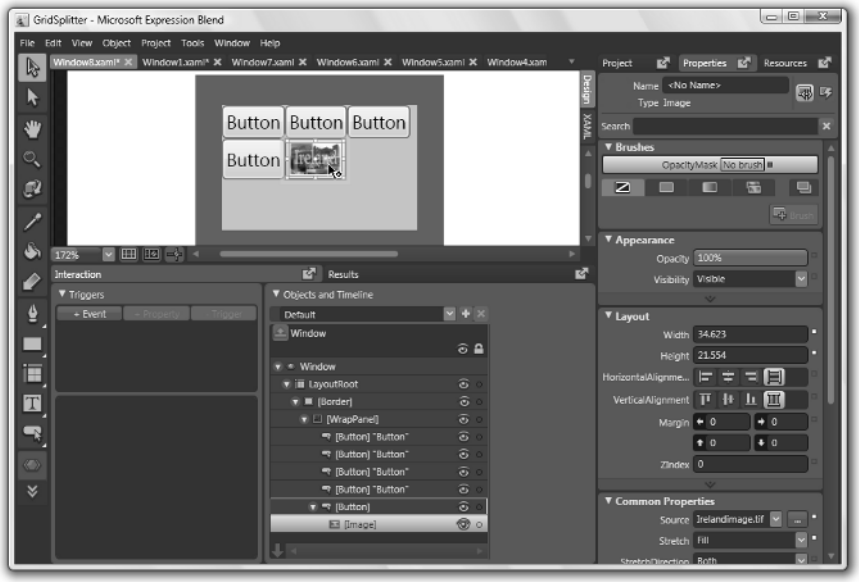
Nesting panels and controls for a hierarchical design

With Blend, you often place the controls for your user interface into a layout panel, but you can also place panels within other panels or controls within other controls. Nested objects, which are called child elements, inherit properties from their parents — for example, they move with the parent when the parent is moved, or they become invisible if the opacity of the parent is set to 0. Some panels or controls can take a single child element, such as a border panel or scroll viewer. Other panels or controls can take many child elements, such as all the panels, except Border, listed in the previous section of this chapter, as well as menus, list boxes, combo boxes, tree views and more. And some controls, such as sliders, progress bars, and grid splitters, accept no child elements.

The Objects list in the Objects and Timeline palette contains all the information about what is nested in what, as shown in Figure 1.12. You can open or close the expanders to explore deeper levels of nesting.

FIGURE 1.12

The image is nested inside a button which is nested in a wrap panel which is nested in a border panel.



When objects are nested, they are still readily available. You can select any particular nested element of your program — to modify it or move it, for example, by double-clicking it several times until the actual element is selected, So if your image is inside a button inside a wrap panel inside a grid panel, as shown in Figure 1.12, you may need to double-click on the image three times and

then single-click it to select the image rather than the panels or the button. Fortunately, this is easier than it sounds, because a large yellow border appears around what is selected to make it obvious what is selected, and double-clicking is fast. When a yellow border appears around an object, then the object is *active*. To make an object inactive, just double-click somewhere else.

TIP **You can also click on the name of the object in the Objects list to select it.**

Nesting makes things much simpler for the designer. You can stack objects, wrap them, place them in sections of a grid, and then move the objects around in their layout panels as a single unit. It makes the task of creating many parts of a user interface simpler by allowing you to compartmentalize objects and controls.

Adding bitmap effects

Blend offers five bitmap effects that you can apply to your controls and other vector objects. They are Bevel, Blur, Drop Shadow, Outer Glow and Emboss. These bitmap effects are a powerful way to add sophisticated imagery to your user interface. For instance, you can give any image the look and feel of a button by giving it a bevel, as shown in the upper left in Figure 1.13. Then you can use Blend's Make Button tool to transform your image into a button. Or you can use the Outer Glow bitmap effect to make your buttons glow when they are pressed or moused over. You can also give your objects drop shadows to add a 3D look to your user interface, and you can add depth to the drop shadow when a tool, for instance, is dragged. You can also use the blur bitmap effect to add a glow to text, as shown in Figure 1.13. The blurred text is behind a text box that contains the same text with a transparent background.

FIGURE 1.13

Bitmap effects applied to images and text.

Using styles and templates

Maintaining consistency in your application is an essential part of making your user interface design highly usable. To aid your efforts in balancing the consistency and interactivity that you need in your interface, you can create and use styles and templates in Blend. You can add animations to styles or you can add property triggers to trigger the change in appearance of a style due to a property change for its object type, such as when the mouse is over the control, or when it is pressed. Blend also offers templates to define the control parts of a control, and how the control parts respond interactively to property changes and events. Using a template, you can define a control that appears radically different from its default WPF control, yet consistently maintains that new look.

You can make styles and templates available as resources within a single window, or within the entire application, or save them in resource dictionaries and make them available to other applications as well. Styles, templates, and resource dictionaries provide you with a way to extend the flexibility of your design, because changing a style or template can result in immediately changing every instance of that style or template.

Linking to data

Using Blend, you can link your controls to different sources of data. For example, you can link your controls to data from RSS feeds, and you can also link to data in databases. Blend allows you to link to two kinds of data formats: Common Language Runtime (CLR) Objects and Extensible Markup Language (XML).

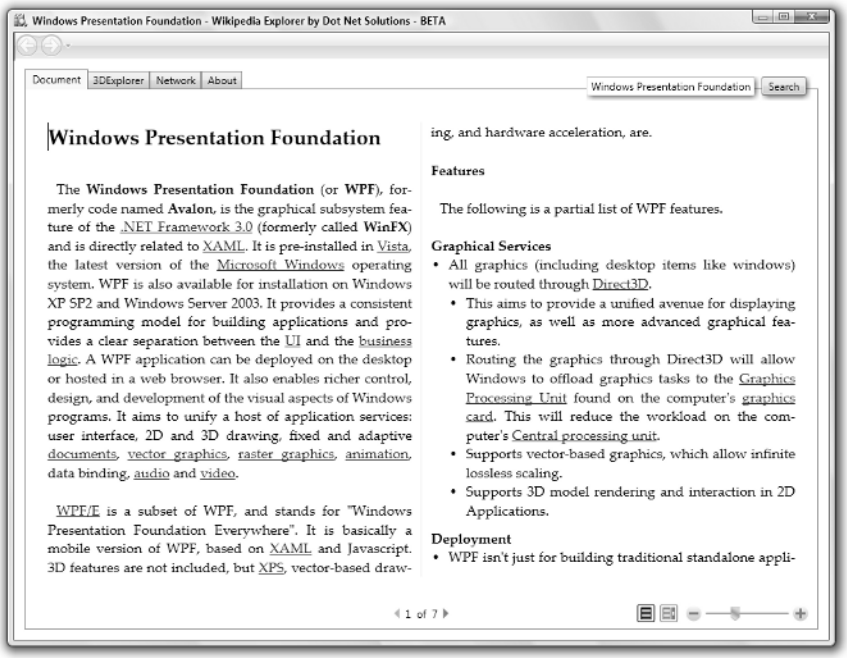
- *CLR Objects* are data items that are usually stored in an ObservableCollection object in the code-behind files. You can read in data from a database into a data table in the code-behind file, and then convert it into an ObservableCollection. Then it can be added as a CLR Object data source in the Data palette of Blend and linked to controls.
- XML, as noted earlier in this chapter, is similar to HTML except that it is extensible. Because it is extensible, the user can create custom tags for data according to certain rules so that the data can be more easily understood by other computer software. These tags can be, for example, the names of fields in your database. You can link to a database and bind controls, such as a list box or combo box, to any field or fields of the database. In Figure 1.14, for example, an XML data source supplies information from Wikipedia.

CROSS-REF **For more information on linking controls to data sources, see Chapter 21.**

ON the WEB **For a great example of how powerful flow documents can be, especially when linked to data sources, go to `http://firstlook.nytimes.com/` and check out the New York Times Reader.**

FIGURE 1.14

The Wikipedia Explorer, developed by Dot Net Solutions, here displays Wikipedia's listing for Windows Presentation Foundation.


Windows Presentation Foundation - Wikipedia Explorer by Dot Net Solutions - BETA
Document
3DExplorer
Network
About
Windows Presentation Foundation
Search
Windows Presentation Foundation
The Windows Presentation Foundation (or WPF), formerly code named Avalon, is the graphical subsystem feature of the .NET Framework 3.0 (formerly called WinFX) and is directly related to XAML. It is pre-installed in Vista, the latest version of the Microsoft Windows operating system. WPF is also available for installation on Windows XP SP2 and Windows Server 2003. It provides a consistent programming model for building applications and provides a clear separation between the UI and the business logic. A WPF application can be deployed on the desktop or hosted in a web browser. It also enables richer control, design, and development of the visual aspects of Windows programs. It aims to unify a host of application services: user interface, 2D and 3D drawing, fixed and adaptive documents, vector graphics, raster graphics, animation, data binding, audio and video.
WPF/E is a subset of WPF, and stands for "Windows Presentation Foundation Everywhere". It is basically a mobile version of WPF, based on XAML and Javascript. 3D features are not included, but XPS, vector-based drawing, and hardware acceleration, are.
Features
The following is a partial list of WPF features.
Graphical Services
All graphics (including desktop items like windows) will be routed through Direct3D.
This aims to provide a unified avenue for displaying graphics, as well as more advanced graphical features.
Routing the graphics through Direct3D will allow Windows to offload graphics tasks to the Graphics Processing Unit found on the computer's graphics card. This will reduce the workload on the computer's Central processing unit.
Supports vector-based graphics, which allow infinite lossless scaling.
Supports 3D model rendering and interaction in 2D Applications.
Deployment
WPF isn't just for building traditional standalone appli-
1 of 7

Increasing Productivity by Increasing Collaboration

In your application, you may want to import sophisticated art that is externally created, and you may also want to connect your user interface design to a back end of complex programming to run the various tools and features that you design into your interface. For this you may want to rely on the work of others. Blend is designed for collaboration both with other designers and with programmers. In large applications, Blend provides the link between the graphics and the back-end programming environment.

Microsoft Expression Design can export (among other things) XAML code, the same format in which Blend stores its data (as described in the section "Including 3D models and animation" earlier in this chapter). So, vector objects created in Design can be imported into Blend easily.

You can also import 3D objects and animation from Zam 3D, which also exports into XAML code. You simply add the xaml file into your project and open it in a document window. Then you can cut and paste its objects into any other window, page or user control in your project. And you can import any 3D object in the popular OBJ format, along with its accompanying MTL (material) file.

The XAML code output by Blend is designed to integrate with projects created with Visual Basic .NET and C#, the popular programming languages from Microsoft. When you create a project in Blend, you can choose between Visual Basic .NET or C# code for your *code-behind files*.

Each document in Blend, such as a window or page, has an accompanying code-behind file which you can use to connect your user interface elements to advanced functionality provided by Visual Basic .NET or C# code. For example, when you want to create code for an object (such as a button, for example) in Blend to make it respond to some event (such as a mouse click, for example), then you type in a name that you want to give to the event handler, and Blend generates the code to create the event handler and opens Visual Studio which pastes the code into the code-behind file and opens to the correct place for you to enter your code into the event handler. If you don't have Visual Studio, you can still open the code-behind file and Blend automatically generates code and places it into the clipboard, which makes it easy for you to paste it into the code-behind file. Then you can type more code into the method that Blend created. (Using Notepad is good for this, although it doesn't offer nearly the functionality of Visual Studio, which provides you with IntelliSense auto-completion, debugging aids, and lots more. But Notepad can be handy for a few lines of code here and there.)

Developers working in Visual Studio and designers working in Blend can work on the same project at the same time. When the project is modified and built in either application, a message appears in the other application, asking whether the project should be updated in that application.

CROSS-REF **For information on importing graphics from Expression Design, see Chapter 7 and Chapter 10.**

For information about using Visual Studio to publish applications and create XAML browser applications for the Web, see Chapter 4.

For more information on modifying and adding code in Blend, see Chapter 20.

For more information on importing 3D art and animation from Zam 3D, see Chapter 12.

Exploring Examples of Innovative User Interfaces

Blend makes it easy to integrate video, audio, animation, 3D, and vector graphics into your user interfaces and thus to take advantage of the extra graphics power of the Windows Presentation Foundation. You may find it useful to merge this use of rich media in interface design with other innovative ideas, such as, for example, user interface ideas taken from the trailblazing work of designer Kai Krause in the 1990s, which we describe next. These ideas may not be useful in every

situation, but they have all been put to work in innovative applications to which Krause contributed, such as Kai's Power Tools, Kai's Power GOO, Kai's Photo Soap, Kai's Power SHOW, LivePicture, Poser, and Bryce.

ON the WEB **You can find Kai's Power Tools at `www.corel.com`, Poser at `www.e-frontier.com`, and Bryce at `www.daz3d.com`.**

Big windows

The idea of using big windows is to immerse the user in the application. For example, if you play a video, you may want to use the entire monitor space while hiding or minimizing the controls. Or if you are in an application, you may want to hide whatever you can to make the window that the user needs as big as possible, as shown in Figure 1.15. Monitors are only so big, and ideally, the user should be able to see only what he or she needs to see for the task at hand. Making use of the entire monitor space without any pixels consumed by a virtual desktop may be advantageous, but you may be able to do more than that by having menus or toolbars disappear when they're not needed. Making use of all the space available can help users keep their attention on what's most important to them.

FIGURE 1.15

The default for the 3D modeling, rendering, and animation program Bryce is to basically take over your whole computer screen as one big window, which works well.

Rooms

Allowing users to work in an environment that focuses just on the single task that they are performing without the distraction of other environments is a big asset. Rooms are big windows laid out for the different features of the application. Rooms can give the feeling of spaciousness in the design, just as many big rooms in a house can make a house feel spacious and opulent. So you may want to give the user lots of rooms.

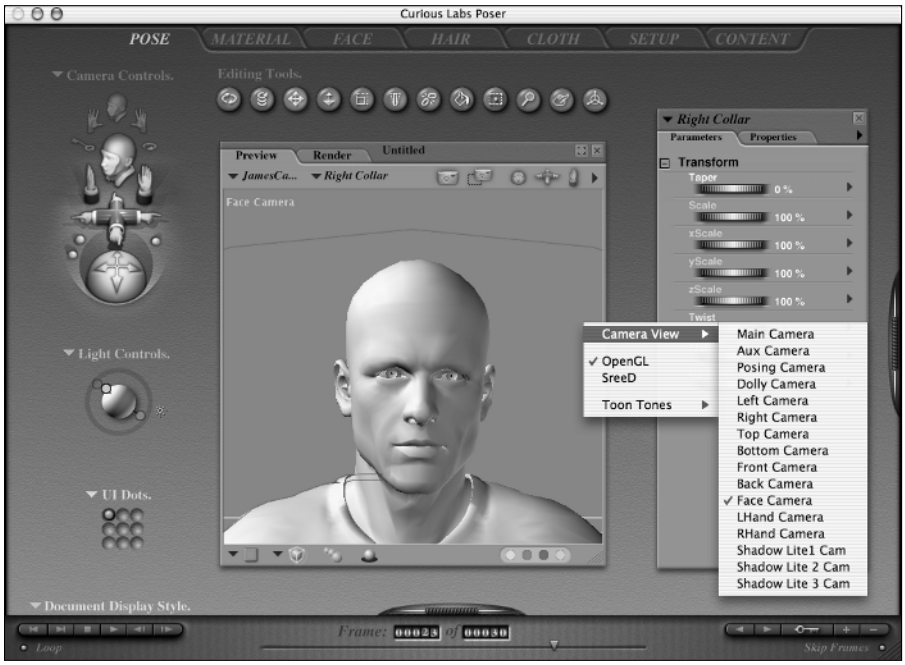
The design of each room should display its functionality. For example, whatever tools you offer specifically for a room should probably be large and prominently displayed so that when the user chooses a room, the functionality of the room is obvious. Perhaps only the tools or objects needed for the user's current task are visible, so that everything in the room is geared to the task at hand.

Minimized interfaces

Offering users the tools they need when they need them may simplify a user interface and increase user productivity and satisfaction. Tools, controls, and menus can pop up when the user needs them (as shown in Figure 1.16) and disappear when the user doesn't need them.

FIGURE 1.16

Pop-up menus like the ones shown here in the 3D modeling program Poser can help minimize the user interface until it's needed.



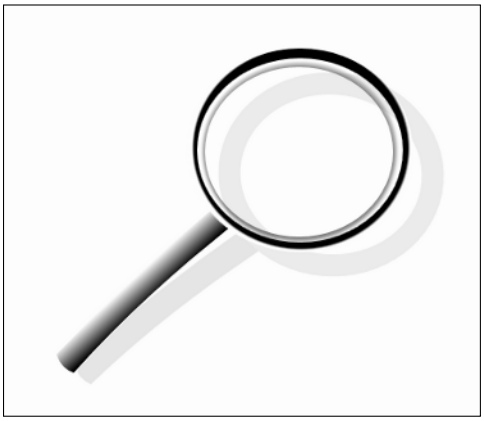
Drag-and-drop desktop tools

Tools that actually look like the tools you may use in real life such as paintbrushes or magnifying glasses, as shown in Figure 1.17, may be useful whenever they can be incorporated into your user interface design, because this adds a natural and familiar element to your application. The tools can be resized to appear large and cast shadows when used. For example, paintbrushes can contain the paints on their tips, which might resize according to the user's preference, and the tips of the paintbrushes might bend as the color is applied. This provides an obvious benefit because users can see the size and color of the brush precisely where they plan to apply it. Users don't need to look over to a distant tool palette to see what they're doing.

Or you might provide your users with drag-and-drop tools that allow them to, for example, drag items that they'd like to buy into a shopping cart, or to drag videos into their "favorites" lists.

FIGURE 1.17

A Zoom tool may become a more appealing and familiar tool when it looks like a magnifying glass and even creates a shadow as it enlarges an area of the screen.



Visibility on demand

In many applications, the number of tools may be more than the user cares to see all at once. Some applications, such as Blend, assign tools to toolbars that can be made visible by choosing them from a menu. Another method might be to have the names of tools grayed out (as shown in Figure 1.18), but appear when you press a key or move the mouse over them. This could let the user see what's available and know where the tools will be when they are needed. Depending on your application, whole groups of toolbars or other controls might appear when a single button is clicked or key is pressed.

FIGURE 1.18

Interface items may be grayed out until they are needed, as menu items are grayed out here in Bryce until the mouse moves over them.

Mouse-overs

Tools can become invisible or small when not used, but when the mouse rolls over the small or invisible tool area, they can pop up and become large and visible and functional once again, as shown in Figure 1.19.

FIGURE 1.19

It may be helpful for tools to become more visible when the mouse passes over them. Here in Bryce, a light glows when the mouse passes over the light.

Memory dots

Offering the user a quick means of switching from one window to another can definitely be a useful addition to the application. For example, the Windows Task Bar shows the applications that are open and allows the user to click on any one of them to quickly go back and forth. This feature, however, does not need to take a large portion of your application. It can be implemented via small buttons or even small dots. For example, buttons could switch you from one window to another within an application.

You may provide several memory dots in your interface design, which when pressed by default take the user to the main screen of your application. From there, whenever the user wants to go to another part of the program, while remembering where he was, he can click another memory dot and start from the main application page again. The original memory dot remembers where the user left off.

Transparencies and shadows

You can give your user interface a 3D look by adding shadows. The magnifying glass in Figure 1.17 casts a shadow, which gives it a floating appearance. Shadows create two effects on the user. First, they may subconsciously give the user the clue that the objects are draggable so that the user

quickly understands how to use them. Second, the shadows may give the application more visual interest.

You may also want to include bitmap effects and opacity masks to create glows and lighting effects in your application design. Glows around objects highlight them, and make them more visible without increasing their size. And you might want to use partial transparency to make tools less visible when they're not being used or not needed.

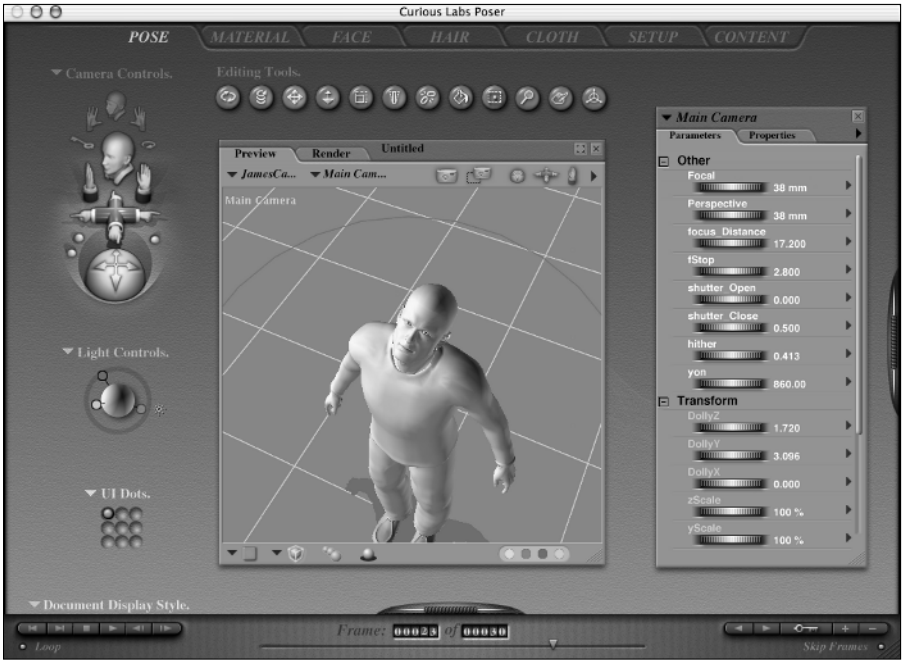
CROSS-REF **See Chapter 7 for more information about transparency, opacity maps, and glows.**

Metawindows, drawers, and scrapbooks

A metawindow is a window into the entire world of your application where you can see the entire functionality of your application at a glance, somewhat like the example shown in Figure 1.20. The user may choose the tools he wants to use or what room he wants to enter, or how he wants to proceed in the application from the metawindow.

FIGURE 1.20

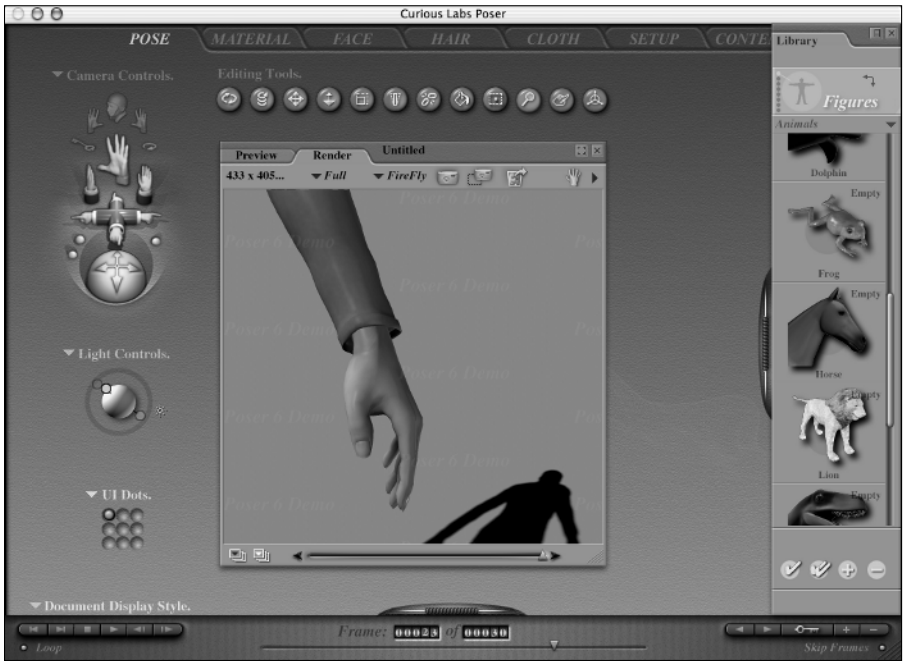
A metawindow may show you the entire functionality of an application at a glance. In this Poser metawindow, many tools are accessible at once.



Drawers are collections of tools or features that can pop out when clicked, as shown in Figure 1.21. Drawers can actually look like drawers or stacked trays, or they can have expanders that indicate that they exist (although that doesn't seem like as much fun).

FIGURE 1.21

You can store all kinds of handy things in drawers. In Poser, the drawers on the right store actors, props, clothing, and more.



Scrapbooks can be a more inviting version of a file folder. A file cabinet full of folders may seem cold and impersonal, but a scrapbook may seem warm and comfortable. In which container would you rather place your images, artwork, or whatever the output of your application is? Of course, if your application creates spreadsheets, maybe they do belong in a file folder.

Even more ideas

The previous user interface ideas are just a few that you may wish to try when using Blend. But don't stop there — Blend makes it easy to push the art of user interface design ahead to an entirely new level, deep into the next generation of advanced user interfaces. You may, for example, want to use Blend to explore:

- *ZUIs* (Zooming User Interfaces) — described by Jef Raskin in his book, *The Humane Interface: New Directions for Designing Interactive Systems* (Addison-Wesley, 2000)
- User interfaces based on layers of movable filters, such as the Magic Lenses proposed at Xerox Parc in the 1990s
- Collaborative, online 3D user interfaces, similar to the Croquet Project, being developed by Alan Kay (one of the designers of the SmallTalk programming language) and others

For more information about the Magic Lenses project at Xerox Parc, point your Web browser to `www2.parc.com/istl/projects/MagicLenses/.`

To learn about the Croquet Project, see `www.opencroquete.org`.

For online demo movies of next generation Web interface and interaction designs, visit Max Kiesler's DesignDemo blog at `www.maxkiesler.com/index.php/designdemo/`.

For late-breaking news on innovative user interfaces of the future, check out the Nooface blog at `www.nooface.com`.

Summary

- You can use Blend to quickly and deeply integrate high quality and highly interactive digital media into your user interfaces — within Blend you can add video, animation, and audio, use 3D tools, import bitmap images, create vector graphics, bind to live data sources, and manipulate text in sophisticated ways to create next-generation Windows applications and Web applications.
- Blend provides you with ready-made WPF controls, and also allows you to radically customize the look of a control while retaining its functions. For example, with Blend you can embed animation, video, and 3D graphics inside your controls.
- The grid, canvas, stack panel, wrap panel, dock panel, and border are the most important panels provided by Blend. Within these panels you can nest objects, controls and other panels, to create a hierarchy of objects that can help you to organize the complex structure of a user interface.
- You can use Blend's styles and templates to maintain consistency in your design as well as to make global changes easily. You can also create and use resource dictionaries to make your styles and templates available to other projects, and to import styles and templates.
- Blend is designed to facilitate collaboration between the artist and the developer — for example, Blend imports a variety of digital media formats, including XAML from Expression Design, Zam 3D and other WPF design applications, and it seamlessly communicates with Visual Studio.
- Blend's extensive support for automatic generation of XAML user interface code and for the advanced capabilities of accelerated 3D graphics and audio-video playback in contemporary computer hardware makes its easy and practical for you to make new breakthroughs in the user experiences provided by applications that you design for Windows and for the Web.